1/16

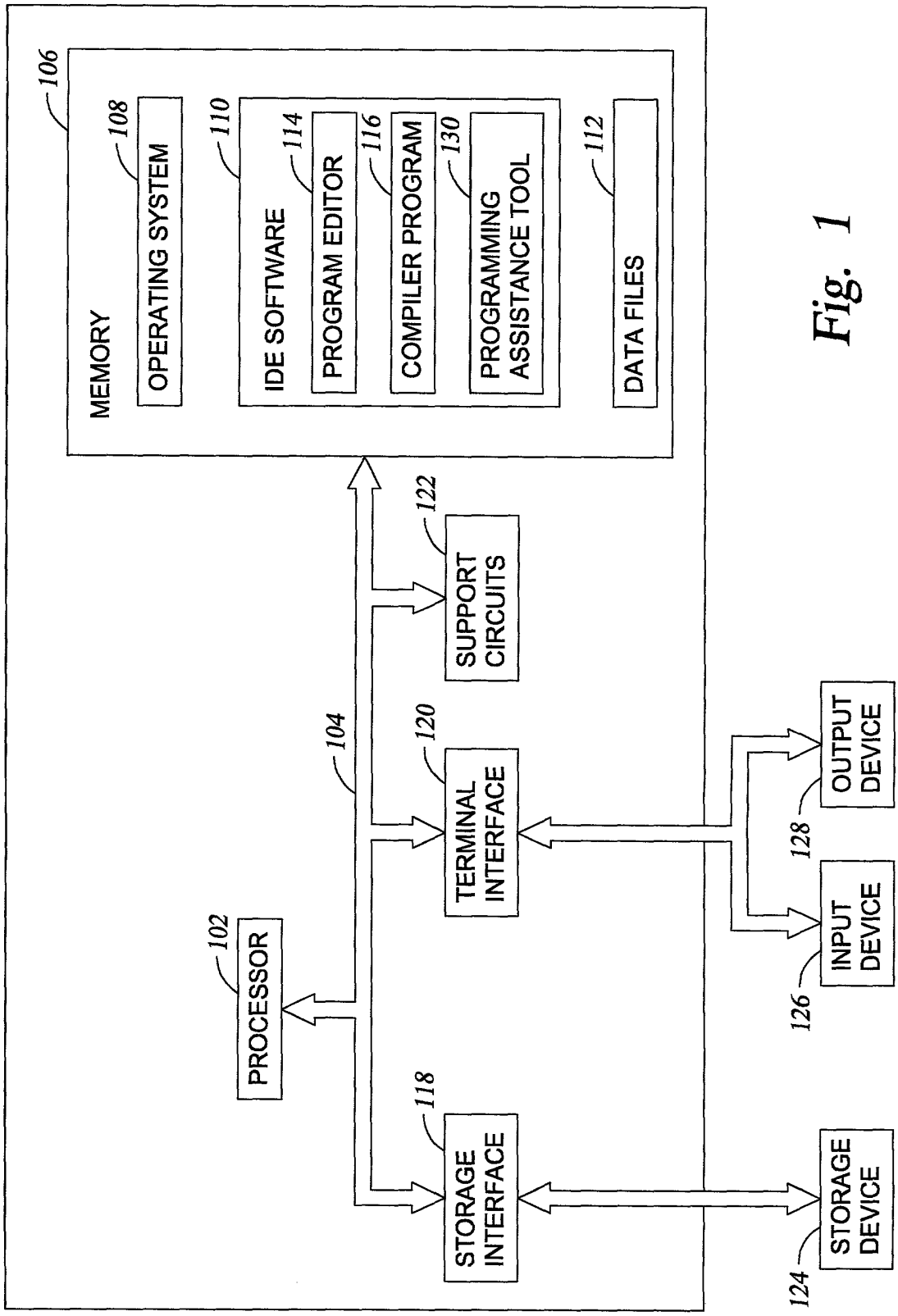

100
MEMORY
106
OPERATING SYSTEM
108
IDE SOFTWARE
110
PROGRAM EDITOR
114
COMPILER PROGRAM
116
PROGRAMMING ASSISTANCE TOOL
130
DATA FILES
112
SUPPORT CIRCUITS
122
TERMINAL INTERFACE
120
104
PROCESSOR
102
STORAGE INTERFACE
118
OUTPUT DEVICE
128
INPUT DEVICE
126
STORAGE DEVICE
124

Fig. 1

METHOD LIST DATA STRUCTURE 200

METHOD DEFINITION	
Example2.method2_2(int)	212
Example2.method2_2(String)	214
Example2.method2_2(char)	216
Example1.method2_2(char)	218
Example1.method2_2(long, String)	220
⋮	222

Fig. 2

VARIABLE TYPE LIST DATA STRUCTURE 300

TYPE	
int	312
String	314
char	316
long	318
⋮	320

Fig. 3

VARIABLE LIST DATA STRUCTURE (AFTER INTERNAL VARIABLES) 400

VARIABLES

int i	412
String s3	414
String this.s2	416
String Example3.s1	418
⋮	420

Fig. 4

LIST OF CLASSES 500

CLASS NAME

java.lang.String	512
java.lang.Interger	514
⋮	516
Example1	518
Example2	520
Example3	522
⋮	524

Fig. 5

VARIABLE LIST DATA STRUCTURE (AFTER EXTERNAL VARIABLES) 600

VARIABLE	
int i	612
String s3	614
String this.s2	616
String Example3.s1	618
String Example2.s1	620
⋮	622

Fig. 6

WEIGHTING DATA STRUCTURE 700

ATTRIBUTE 702	WEIGHT 704	
MOST RECENTLY CREATED/MODIFIED	+30	712
LOCAL VARIABLE	+10	714
FROM CLASS HEIRARCHY	+5	716
EXTERNAL VARIABLE	-5	718
USED PREVIOUSLY	-10	720
INITIALIZED	-20	722
⋮	⋮	724

Fig. 7

VARIABLE LIST DATA STRUCTURE (AFTER RANKING) 800

VARIABLE (802)	(804)	
int i	2	812
String s3	45	814
String this.s2	22	816
String Example3.s1	10	818
String Example2.s1	0	820
⋮	⋮	

Fig. 8

900

VARIABLE (902)	RANKING (904)	
Sting S3	90%	914
int i	90%	916
String this.s2	45%	918
String Example3.s1	20%	920
String <new>	5%	921
int #	5%	922
Char "c"	5%	924
long #	5%	926
String Example2.s1	1%	928

Fig. 9

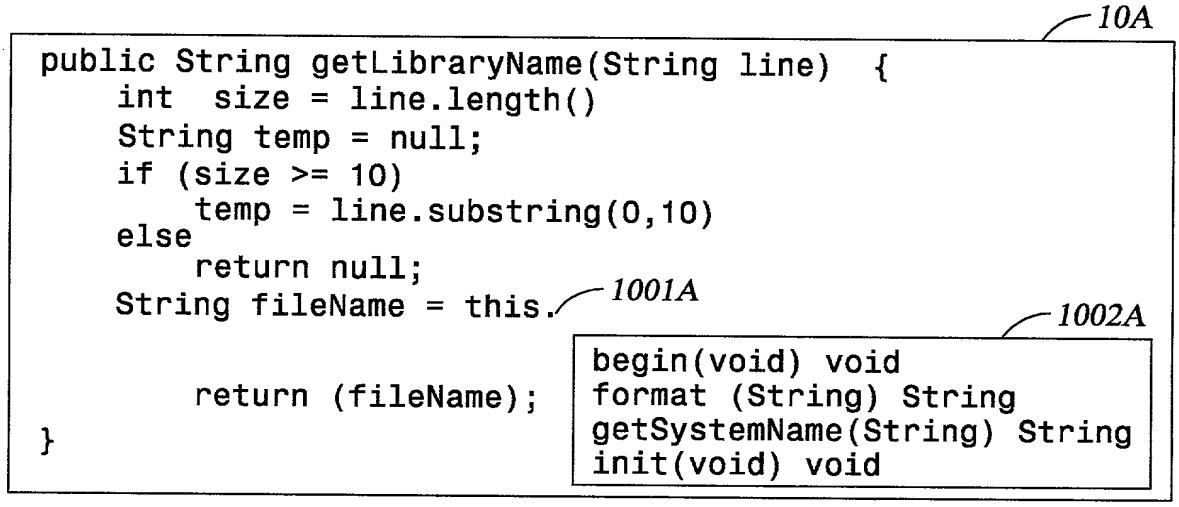


Fig. 10A

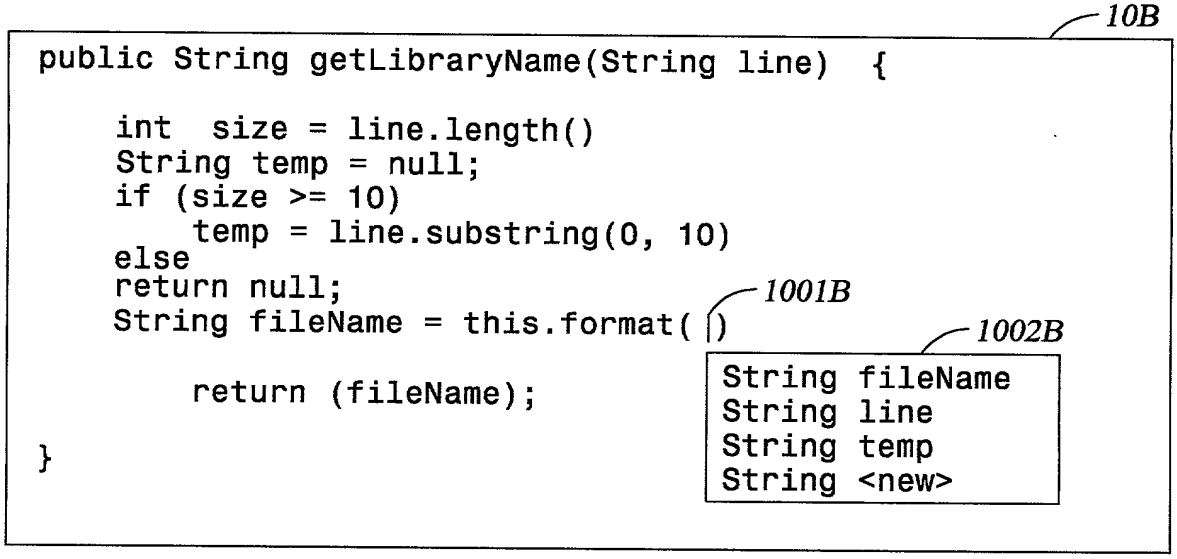


Fig. 10B

Fig. 11A

```
public class Example1 {
   public void method2_2(char c) {
      return;
   }
   public void method2_2(long l, String s) { // overloaded method
      return;
   }
}

public class Example2 extends Example1 {

   public static String s1 = "hello"; // class variable
   public String s2 = "there"; // instance variable
   public String s3 = "don't touch"; // variable inaccessable externally

   public static void method2_1() {// class method
      return;
   }
   public static void method2_2(int i){ // instance method
      String s= "jump"; // local variable, instance of the String class
      Example2.method2_1(); // calling a class method on a class object
      s.length(); // calling an instance method
   }
   public void method2_2(String s){ // overloaded method
      return;
   }
```

```
    public void method2_2(char c){ // overloaded and overriden method
      return;
    }
}

    public class Example3 {

    private static Strig s1 = "class";
    private String s2 = "instance"

    public void method3_1(int i) {
      String s3 = "local";

      Example2 ex2 = new Example2();
      ex2.method2_2( ) 1001C
```

90%	String s3
90%	int i
45%	String this.s2
20%	String Example3.s1
5%	String <new>
5%	int #
5%	char 'c'
5%	long #
1%	String Example2.s1

1002C

Fig. 11B

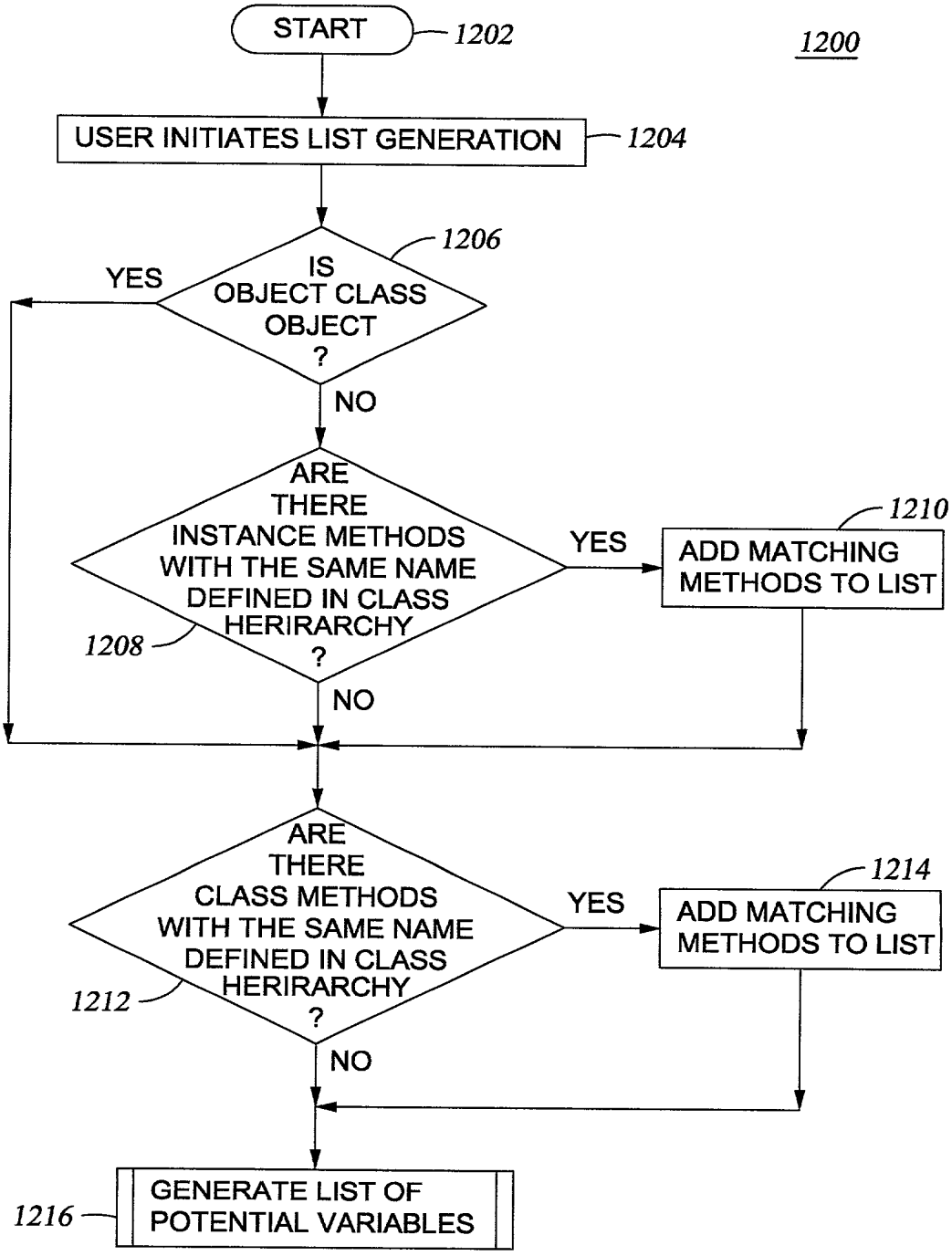

START
1202
1200
USER INITIATES LIST GENERATION
1204
IS OBJECT CLASS OBJECT ?
1206
YES
NO
ARE THERE INSTANCE METHODS WITH THE SAME NAME DEFINED IN CLASS HERIRARCHY ?
1208
YES
ADD MATCHING METHODS TO LIST
1210
NO
ARE THERE CLASS METHODS WITH THE SAME NAME DEFINED IN CLASS HERIRARCHY ?
1212
YES
ADD MATCHING METHODS TO LIST
1214
NO
GENERATE LIST OF POTENTIAL VARIABLES
1216

Fig. 12

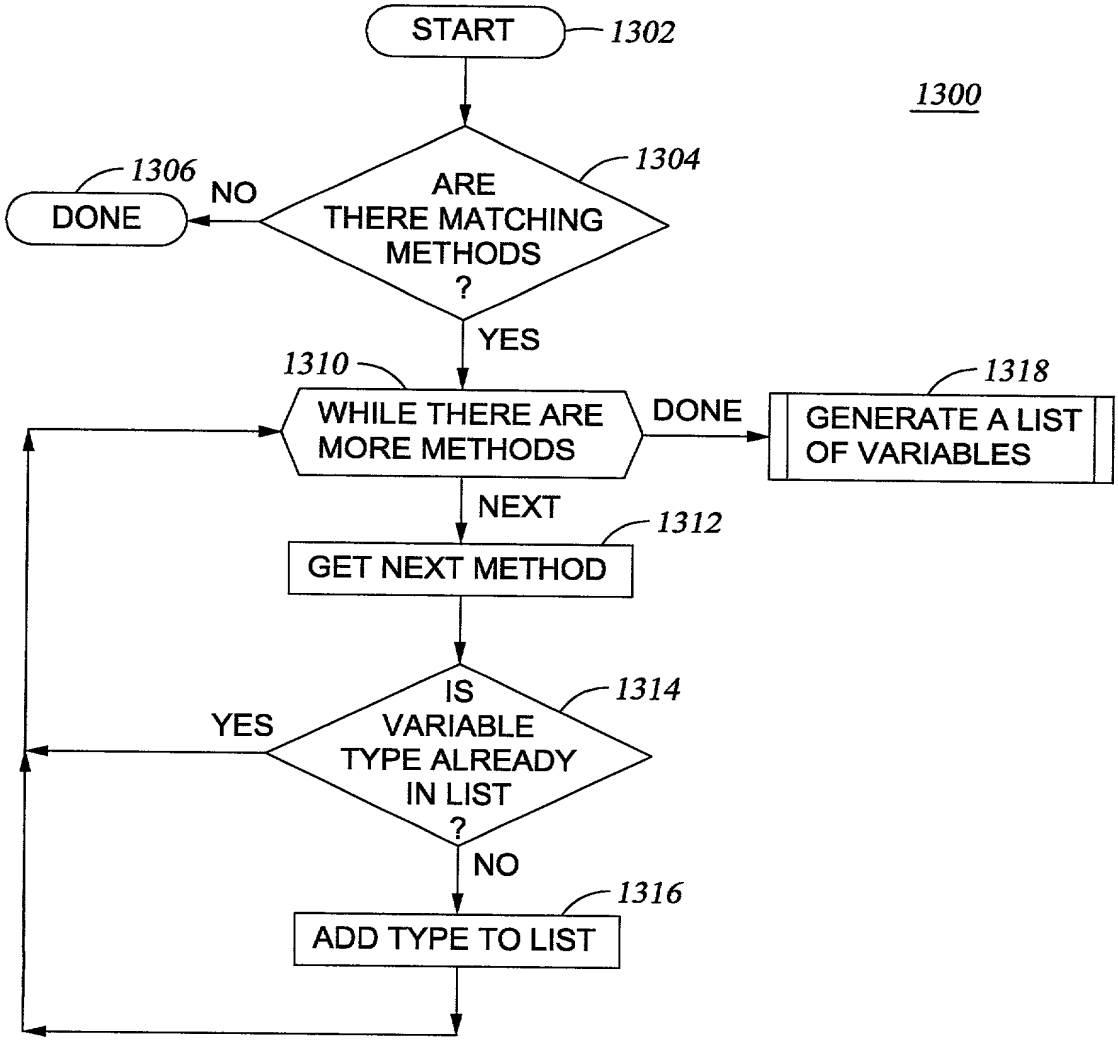
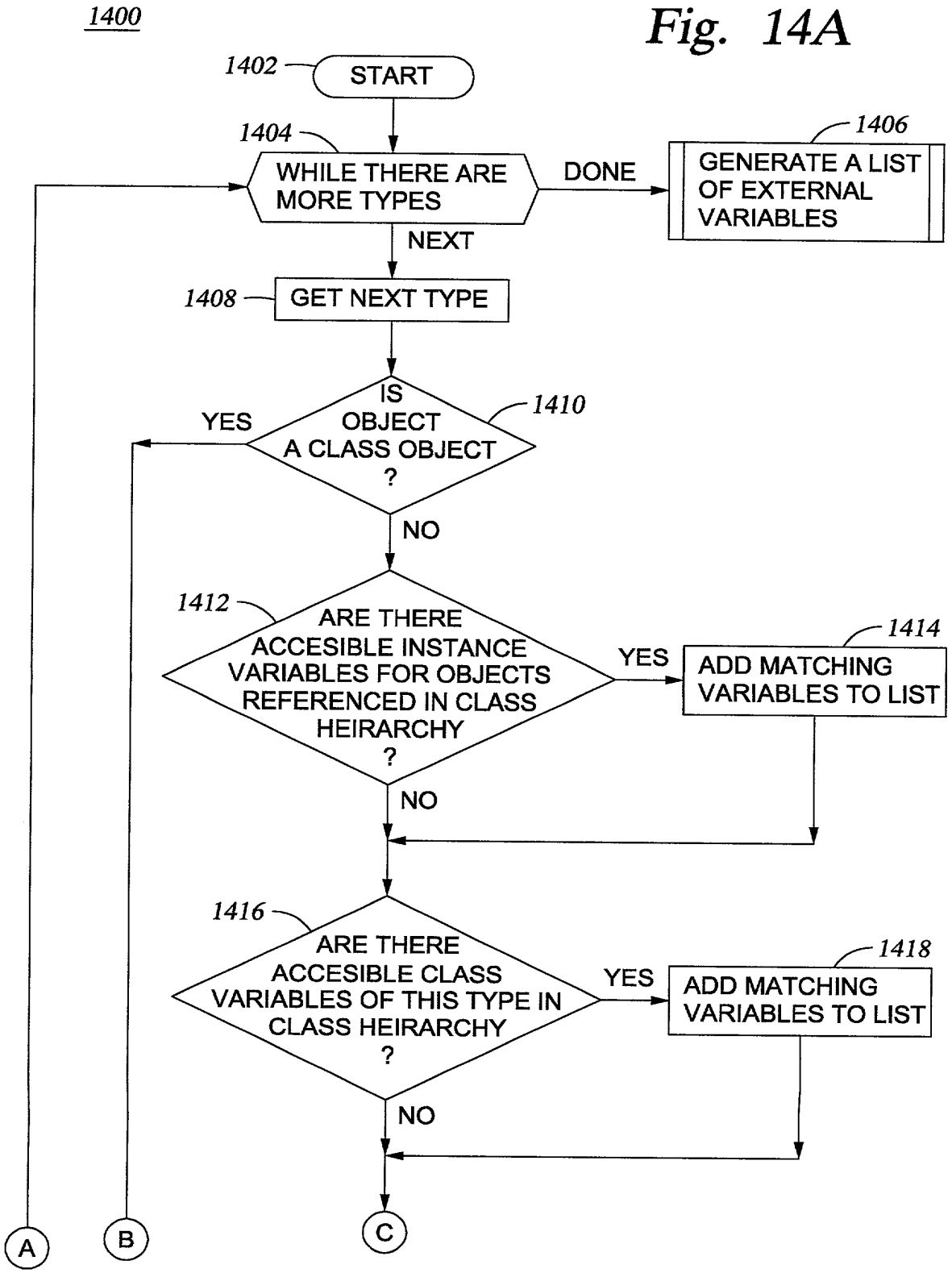

START
1302
1300
1306
NO
DONE
ARE THERE MATCHING METHODS ?
1304
YES
1310
WHILE THERE ARE MORE METHODS
DONE
1318
GENERATE A LIST OF VARIABLES
NEXT
1312
GET NEXT METHOD
IS VARIABLE TYPE ALREADY IN LIST ?
1314
YES
NO
1316
ADD TYPE TO LIST

Fig. 13

Fig. 14A


1400
1402
START
1404
WHILE THERE ARE MORE TYPES
DONE
1406
GENERATE A LIST OF EXTERNAL VARIABLES
NEXT
1408
GET NEXT TYPE
IS OBJECT A CLASS OBJECT ?
1410
YES
NO
1412
ARE THERE ACCESIBLE INSTANCE VARIABLES FOR OBJECTS REFERENCED IN CLASS HEIRARCHY ?
YES
1414
ADD MATCHING VARIABLES TO LIST
NO
1416
ARE THERE ACCESIBLE CLASS VARIABLES OF THIS TYPE IN CLASS HEIRARCHY ?
YES
1418
ADD MATCHING VARIABLES TO LIST
NO
A
B
C

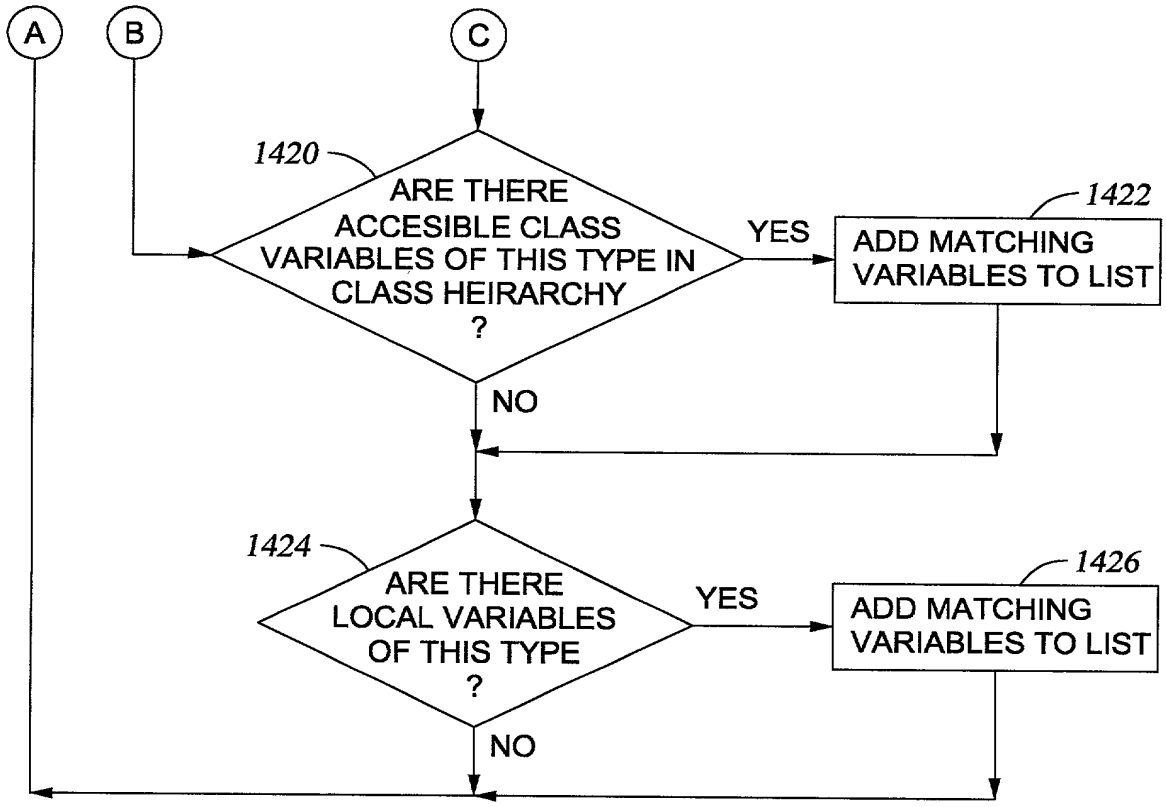

A
B
C
1420
ARE THERE ACCESIBLE CLASS VARIABLES OF THIS TYPE IN CLASS HEIRARCHY ?
YES
1422
ADD MATCHING VARIABLES TO LIST
NO
1424
ARE THERE LOCAL VARIABLES OF THIS TYPE ?
YES
1426
ADD MATCHING VARIABLES TO LIST
NO

Fig. 14B

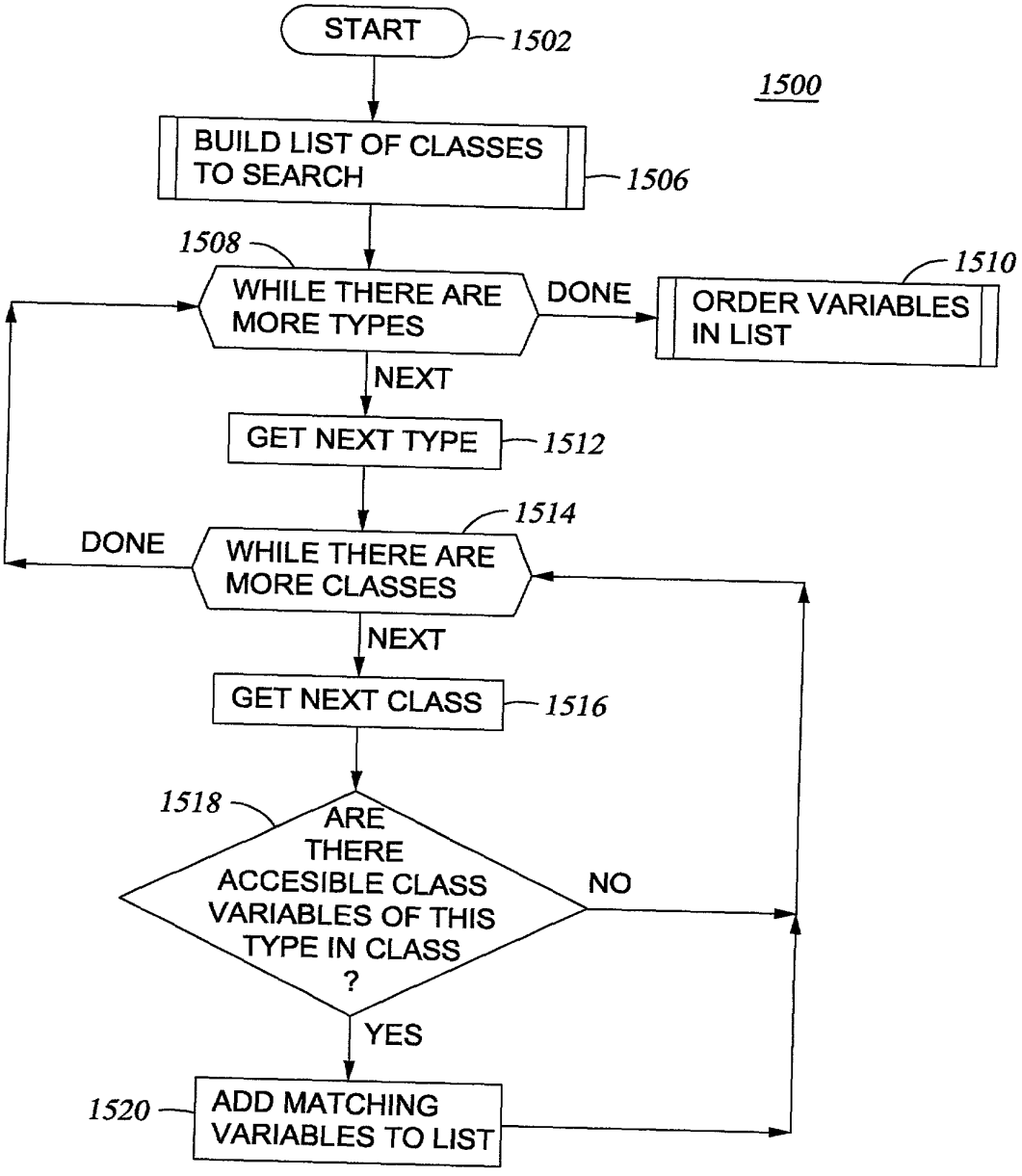

START
1502
1500
BUILD LIST OF CLASSES TO SEARCH
1506
1508
WHILE THERE ARE MORE TYPES
DONE
1510
ORDER VARIABLES IN LIST
NEXT
GET NEXT TYPE
1512
1514
DONE
WHILE THERE ARE MORE CLASSES
NEXT
GET NEXT CLASS
1516
1518
ARE THERE ACCESIBLE CLASS VARIABLES OF THIS TYPE IN CLASS ?
NO
YES
1520
ADD MATCHING VARIABLES TO LIST

Fig. 15

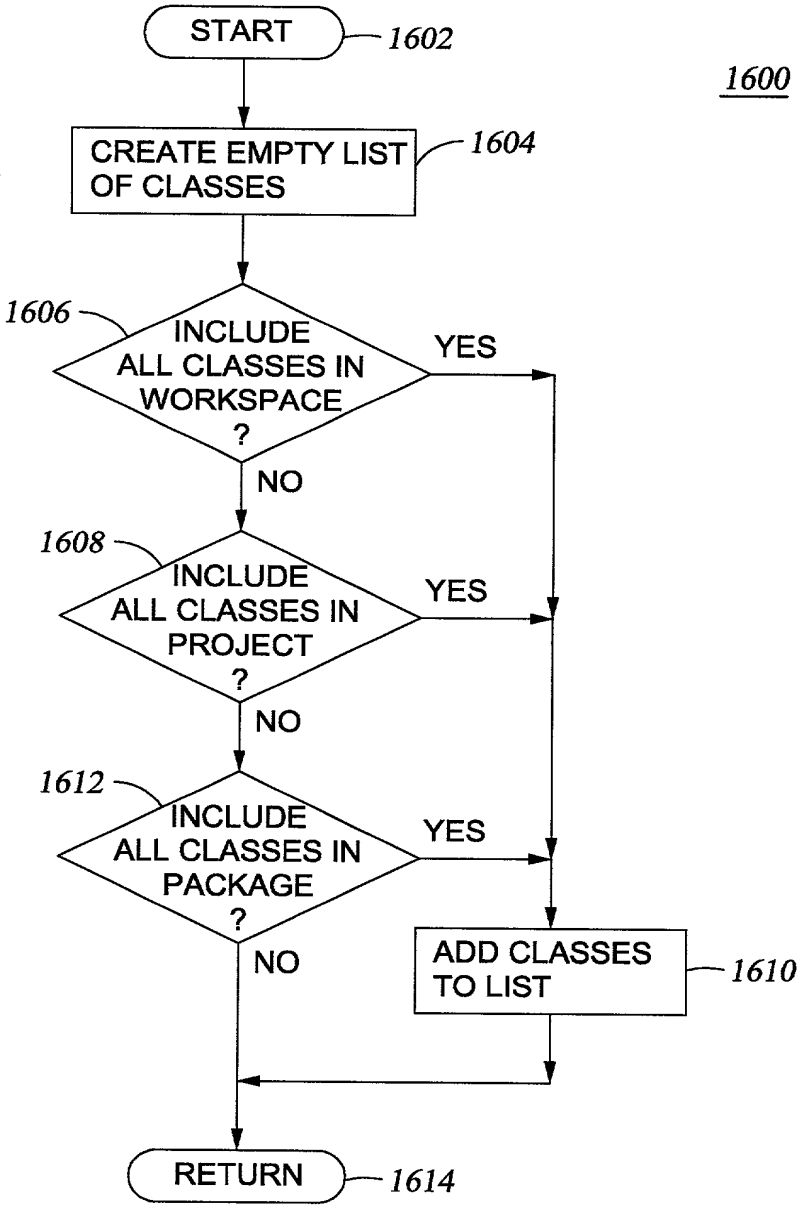

START
1602
1600
CREATE EMPTY LIST OF CLASSES
1604
1606
INCLUDE ALL CLASSES IN WORKSPACE ?
YES
NO
1608
INCLUDE ALL CLASSES IN PROJECT ?
YES
NO
1612
INCLUDE ALL CLASSES IN PACKAGE ?
YES
NO
ADD CLASSES TO LIST
1610
RETURN
1614

Fig. 16

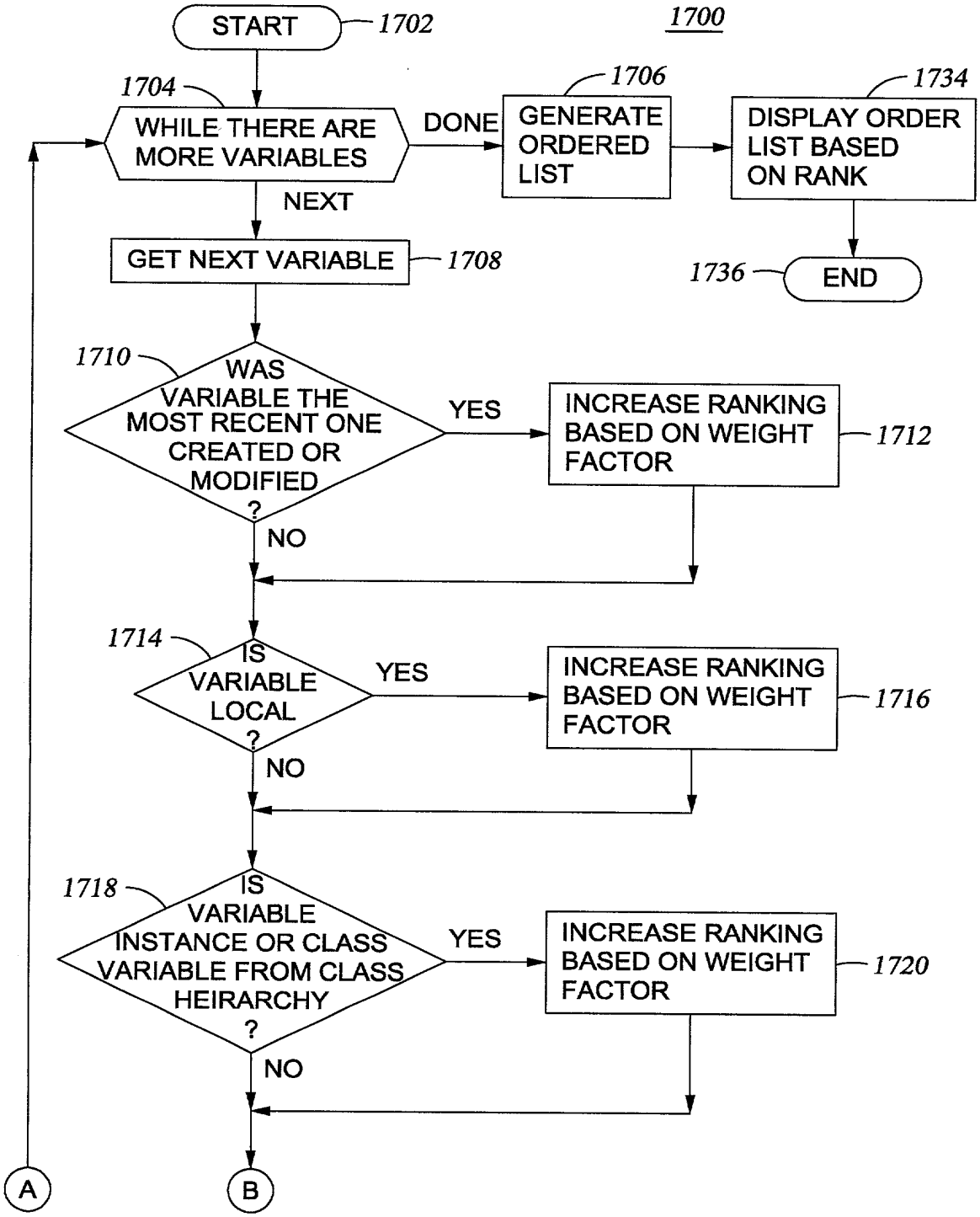

1700
START
1702
1704
WHILE THERE ARE MORE VARIABLES
DONE
1706
GENERATE ORDERED LIST
1734
DISPLAY ORDER LIST BASED ON RANK
NEXT
GET NEXT VARIABLE
1708
1736
END
1710
WAS VARIABLE THE MOST RECENT ONE CREATED OR MODIFIED ?
YES
INCREASE RANKING BASED ON WEIGHT FACTOR
1712
NO
1714
IS VARIABLE LOCAL ?
YES
INCREASE RANKING BASED ON WEIGHT FACTOR
1716
NO
1718
IS VARIABLE INSTANCE OR CLASS VARIABLE FROM CLASS HEIRARCHY ?
YES
INCREASE RANKING BASED ON WEIGHT FACTOR
1720
NO
A
B

Fig. 17A

16/16

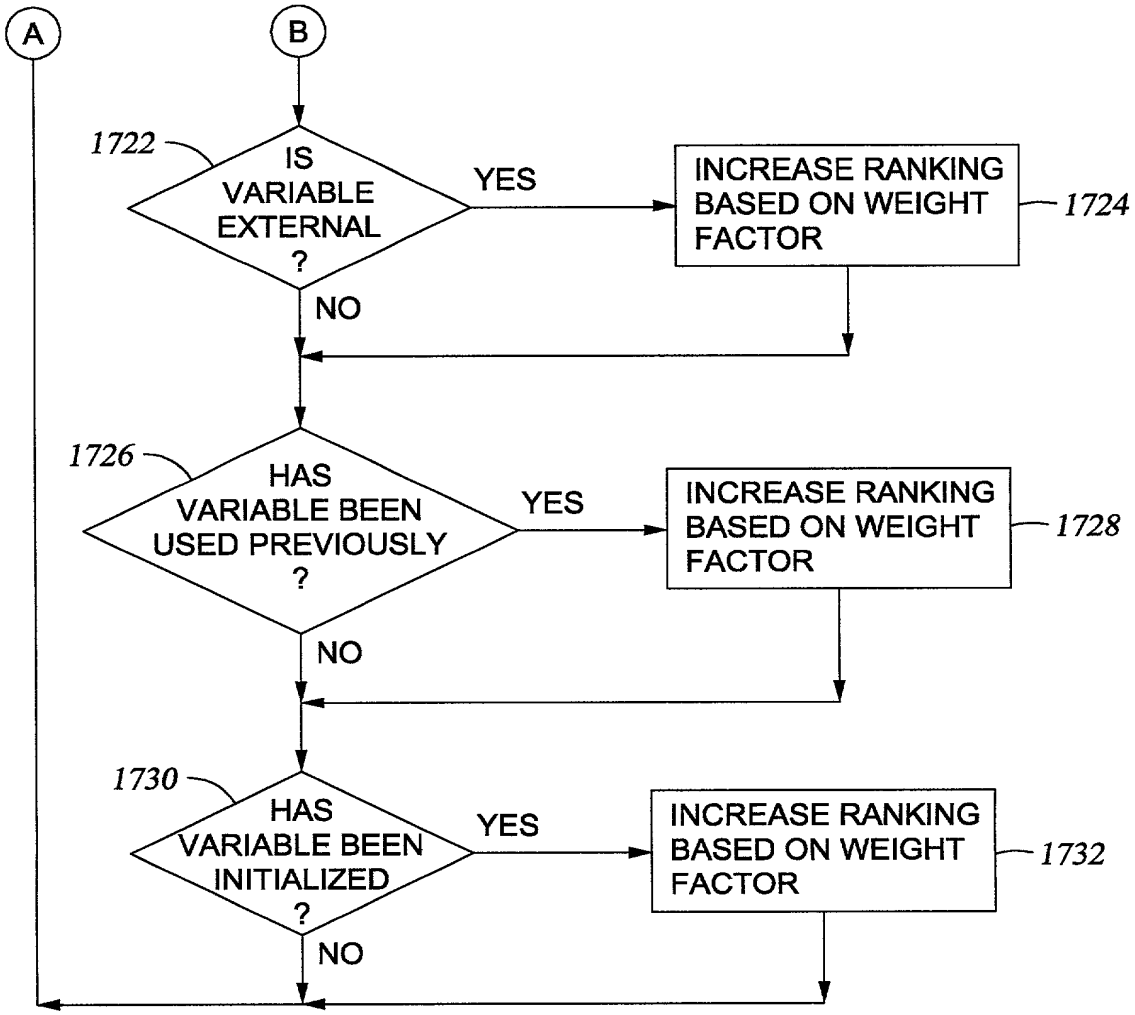

A
B
1722
IS VARIABLE EXTERNAL ?
YES
INCREASE RANKING BASED ON WEIGHT FACTOR
1724
NO
1726
HAS VARIABLE BEEN USED PREVIOUSLY ?
YES
INCREASE RANKING BASED ON WEIGHT FACTOR
1728
NO
1730
HAS VARIABLE BEEN INITIALIZED ?
YES
INCREASE RANKING BASED ON WEIGHT FACTOR
1732
NO

Fig. 17B